

What is claimed is:

1. A computer system comprising:
a preloader arranged to,
5 determine whether a bytecode makes an active reference to a class
which requires an execution of a static initializer,
determine if the class has a superclass which requires the execution of
the static initializer, wherein the preloader produces a source file;

10 a compiler coupled to the preloader arranged to accept the source file as input
and produce an object file; and

a virtual machine coupled to the compiler arranged to execute the object file.

15 2. A computer system according to claim 1 wherein the preloader is
further arranged to:
rewrite the bytecode to a new bytecode which indicates that at least one of the
class and the superclass requires execution of the static initializer when it is
determined that the bytecode makes the active reference to the class which requires
20 the execution of the static initializer.

3. A computer system according to claim 1 wherein the preloader is
further arranged to:
rewrite the bytecode to a new bytecode which explicitly indicates that at least
25 one of the class and the superclass requires execution of the static initializer when it is
determined that the bytecode makes the active reference to the class which requires
the execution of the static initializer.

4. A computer system according to claim 1 wherein the preloader is
30 further arranged to:
rewrite the bytecode to a new bytecode which indicates that at least one of the
class and the superclass requires execution of the static initializer when it is

determined that the bytecode makes the active reference to the class which has the superclass which requires the execution of the static initializer.

5. A computer system according to claim 1 wherein the preloader is further arranged to:

rewrite the bytecode to a new bytecode which explicitly indicates that at least one of the class and the superclass requires execution of the static initializer when it is determined that the bytecode makes the active reference to the class which has the superclass which requires the execution of the static initializer.

6. A computer system comprising:

a bytecode rewriter arranged to,

determine whether a bytecode is associated with a scalar field or an object reference field,

rewrite the bytecode to identify the bytecode as being associated with the scalar field when the bytecode is associated with the scalar field,

rewrite the bytecode to identify the bytecode as being associated with the object reference field when the bytecode is associated with the object reference field, wherein the bytecode rewriter is associated with producing a source file;

a compiler arranged to accept the source file as input and produce an object file; and

a virtual machine arranged to execute the object file.

7. In a computer system having a preloader coupled to a compiler and a virtual machine, a method for rewriting bytecodes to minimize runtime checks, comprising:

by the preloader,

determining whether a bytecode makes an active reference to a class which requires an execution of a static initializer;

determining if the class has a superclass which requires the execution of the static initializer, wherein the preloader produces a source file;

accepting the source file as input and produce an object file by the compiler; and

executing the object file by the virtual machine.

8. A method according to claim 7 further comprising:

rewriting the bytecode to a new bytecode, by the preloader, which indicates that at least one of the class and the superclass requires execution of the static initializer when it is determined that the bytecode makes the active reference to the class which requires the execution of the static initializer.

9. A method according to claim 7, further comprising:

rewriting the bytecode to a new bytecode, by the preloader, which explicitly indicates that at least one of the class and the superclass requires execution of the static initializer when it is determined that the bytecode makes the active reference to the class which requires the execution of the static initializer.

10. A method according to claim 7 further comprising:

rewriting the bytecode to a new bytecode, by the preloader, which indicates that at least one of the class and the superclass requires execution of the static initializer when it is determined that the bytecode makes the active reference to the class which has the superclass which requires the execution of the static initializer.

11. A method according to claim 7, further comprising:

rewriting the bytecode to a new bytecode, by the preloader, which explicitly indicates that at least one of the class and the superclass requires execution of the static initializer when it is determined that the bytecode makes the active reference to the class which has the superclass which requires the execution of the static initializer.

5

0944759-042404
104240-65214860